ReDS

Systèmes embarqués

D.Rossier

Manuel IEM (v0.4)

Manuel d'utilisation pour les laboratoires IEM

Auteur: G. Boutillier / C. Bardet / L. Sambuc

Objectif

Cette documentation donne de brèves explications sur l'utilisation des différents outils, environnements, protocoles utilisés dans le laboratoire de sytèmes embarqués. Ce document est un résumé, il ne remplace en aucun cas les différentes documentations fournies au début du laboratoire.

1. Carte de laboratoire EMB

La carte de laboratoire EMB est équipée d'une carte Freescale CSB535FS constituée d'un microcontrôleur i.MX21 (ARM926EJ-S). Elle possède 64Mo de mémoire SDRAM et 8Mo de mémoire FLASH. Une carte périphérique additionnelle est connectée à la carte CPU, elle possède un écran tactile, différents ports (UART, Ethernet, USB host/device), des connecteurs pour l'utilisation d'une carte Compact Flash et SD-Card, des entrées/sorties audio, des boutons et des LEDs disponibles à travers un périphériques I^2C, etc. Pour les laboratoires une carte interface a été rajoutée, elle possède 2 boutons et 2 LEDs directement connectés sur des GPIOs du i.MX21, ainsi que différents systèmes pour la génération d'horloges.

L'adresse IP de la carte EMB est **10.0.0.2**.

Connexion à la carte EMB depuis une machine Compaq:

- Alimenter la carte en 12V avec des fiches bananes depuis une alimentation du laboratoire
- Connexion du port série de la machine hôte à la cible
- Connexion Ethernet de la machine hôte à la cible (carte réseau **2**, câble Ethernet croisée rouge)

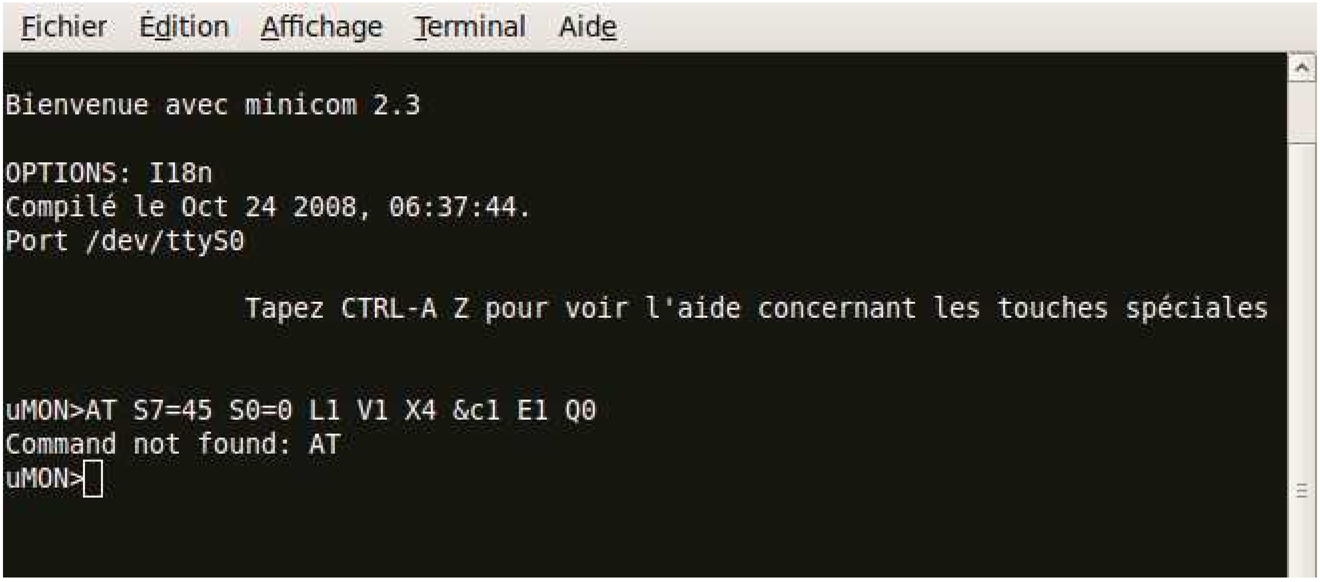
2. Minicom

Minicom est un programme utilisé dans ce laboratoire pour la connexion via le port série au moniteur ou au Linux embarqué sur la carte CSB535 depuis le Linux de la machine hôte.

Connexion:

Taper dans une console sur la machine hôte :

```
$ minicom
```



Vous êtes maintenant connecté à votre cible!

3. Micro-monitor (uMON)

Micro-monitor (ou micro-moniteur) est le moniteur installé sur la carte CSB535. Il est stocké en flash mais exécuté en RAM. Il possède des commandes qui lui sont propres. Les commandes spécifiques à la gestion du système de fichiers TFS - *Tiny File System* - en *flash* sont similaires à celles de Linux. Pour ces dernières, il suffit de débuter votre commande par *tfs* suivi de la commande standard linux. Exemple: ***tfs ls***

4. Machine hôte Compaq

Les machines Compaq ont été configurées afin de disposer de tous les outils, environnements, services dont vous avez besoin pour le laboratoire de systèmes embarqués. Elles possèdent deux cartes réseau Ethernet : une carte pour la connexion au réseau local EMB, la deuxième pour la connexion à votre cible. L'adresse IP de cette deuxième carte réseau est **10.0.0.1**.

Pour transférer un fichier en flash sur la cible (scripts, exécutable, etc), vous devez taper la commande suivante sur une console de la machine hôte :

$ tftp 10.0.0.2

tftp> **put** *nom_fichier_hôte nom_fichier_cible*

Pour transférer un fichier de la cible à la machine hôte, taper dans une console hôte :

$ tftp 10.0.0.2

tftp> **get** *nom_fichier_cible nom_fichier_hôte*

Note importante : Si vous transférer un fichier exécutable, vous devez l'indiquer en ajoutant *,E* après le *nom_fichier_cible. Pour un script ajouter un ,e*

Exemple:

$ tftp 10.0.0.2

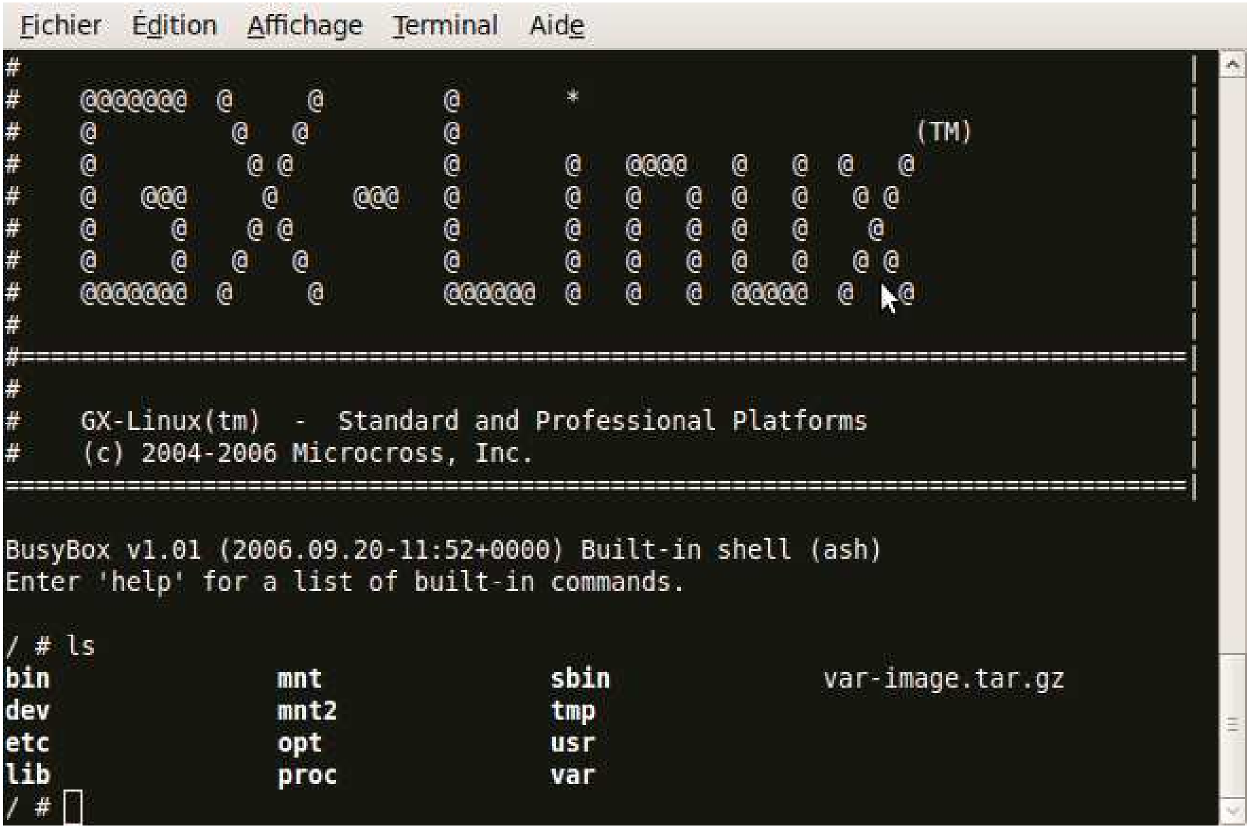
tftp> put helloworld helloworld**,E**

5. Linux embarqué

Pour démarrer un Linux embarqué, il faut avoir en flash le noyau Linux embarqué (*zImage*), l'arborescence racine compressé (*romfs.img*) et le script de démarrage *startlinux*. Tous ces composants ont été transférés à l’avance sur les cartes de laboratoires à l’aide des commandes précédentes.

1. Pour démarrer Linux, lancer depuis le micro-monitor le script *startlinux*.

uMON : startlinux



Vous avez maintenant démarré linux sous la cible, tester en essayant la commande ***ls***.

2. Transférer une application s'exécutant sous Linux embarqué sur la cible :

Machine hôte : Copier l'exécutable que vous désirez transférer sur la cible dans le répertoire /share ou dans le répertoire *share_freescale* placé sur le bureau.

Cible : Aller dans le répertoire /var, créer un répertoire portant votre nom. Depuis ce même répertoire taper dans la console :

$ tftp –g –r *nom_fichier* 10.0.0.1

Puis changer les droits de votre fichier pour pouvoir l'exécuter :

$ chmod 777 *nom_fichie*r

6. Qemu

Qemu est un programme permettant d'émuler un autre environnement. Ce programme est très intéressant pour pouvoir tester des aplications embarquées sans pour autant avoir besoin de charger les exécutables sur une carte. Il est également possible d'émuller un autre environnemnt depuis Linux (par ex Windows) ce qui peut être pratique pour utiliser certain programme.

Sur les machines compaq Qemu a été compilé pour les systems ARM.

Qemu s'exécute avec la commade suivante:

$ qemu-system-arm *[suivit de divers paramètres]*

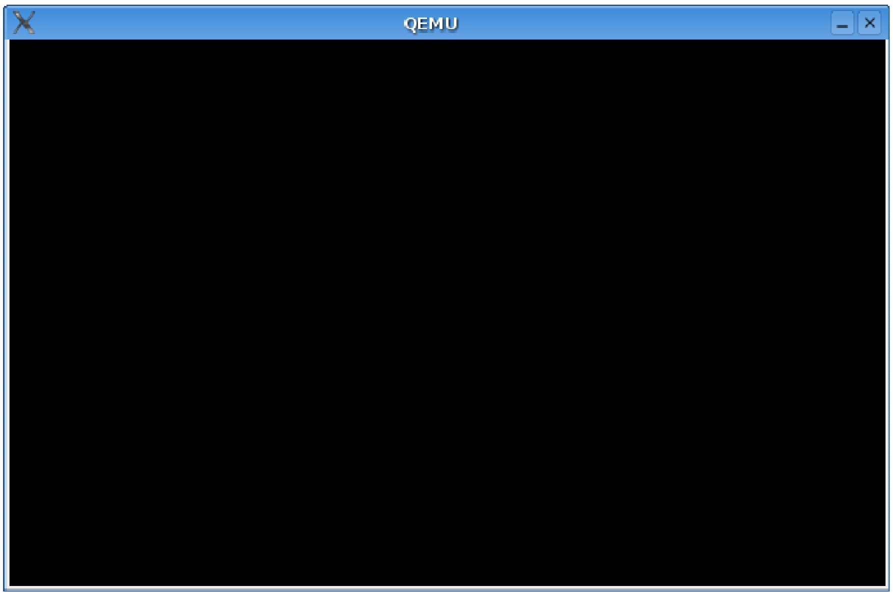
Pour lister les paramètres:

$ qemu-system-arm –help

ou simplement

$ qemu-sytem-arm

Qemu s'ouvre dans une nouvelle fenêtre:

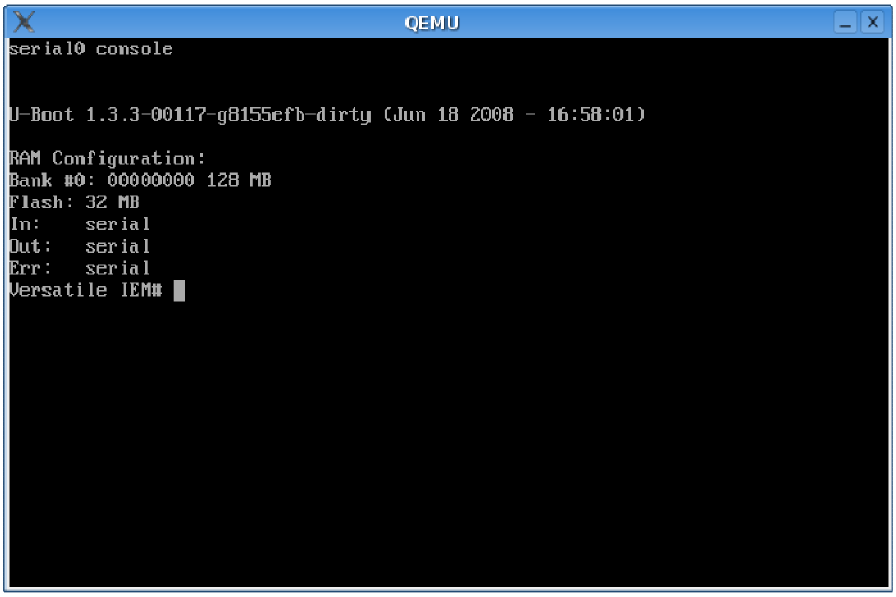


Qemu fonctionne sur quatre fenêtres différentes:

La fenêtre d'accueil (celle ci dessus) contient l'environement émulé (Ctrl-Alt 1). Pour changer de mode, il faut utiliser les touches Ctrl-Alt + touche 1 à 4:

Ctrl-Alt 2: Affiche les informations du "monitor"

Ctrl-Alt 3: Commute sur la console serie (voir ci dessous)

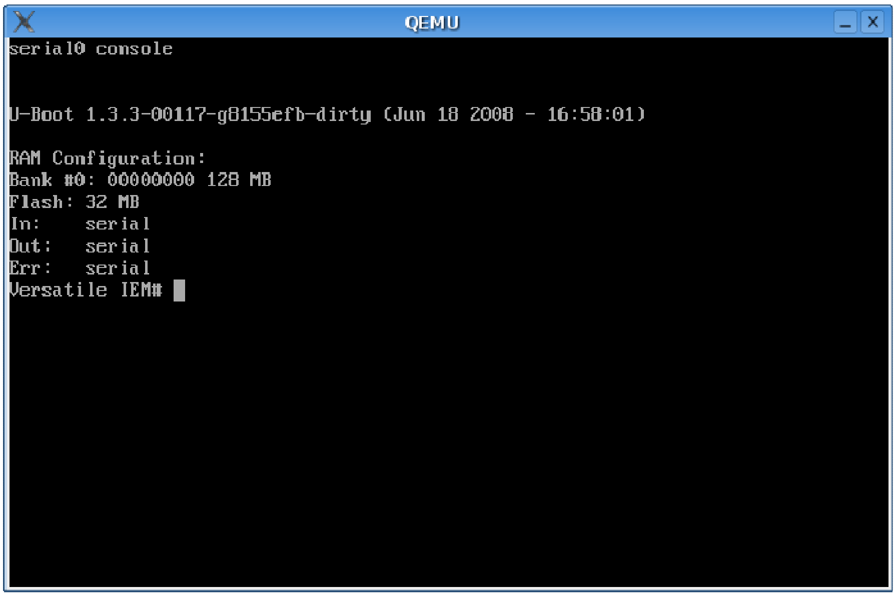


Ctrl-Alt 4: Commute sur la console parallèle

Ctrl-Alt: Sort du system émulé et redonne la mains a l'OS principal (Linux-Ubuntu) ainsi que les périphériques ci-rapportant (par ex: la souris) .

7. U-boot

U-boot.bin est paramétré dans la commande qemu-system-arm. On le retrouve dans la console série de Qemu voir ci-dessous. Versatile IEM est une émmulation d'une machine ARM 926EJS.



Voici quelque commande de base:

- help (ou ?): liste et explique toute les commandes disponibles de U-boot

  # **help printenv** (explique la commande printenv)

- printenv: liste les variables d'environnement ainsi que leur valeurs.

- setenv: Ajoute ou modifie une variable d'environnement (sert a tranférer un prog)

  # **setenv *nom_variable* tftp 0x*adresse-de-destination-du-prog* /adresse/source/prog**

- run: charge un programme défini par une variable d'environnement

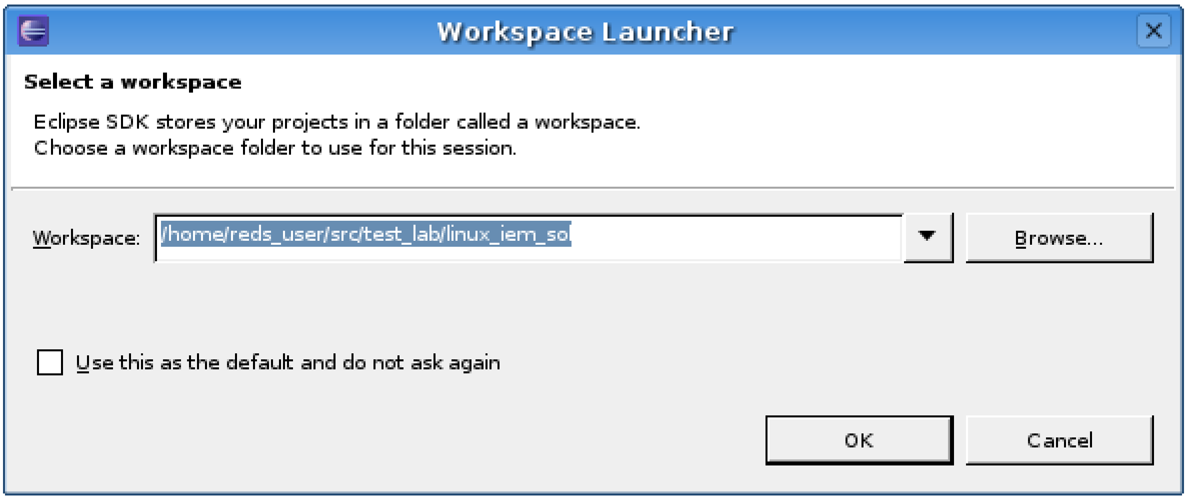
  # **run nom_variable**

- go: démmare l'application a l'adresse 0x????

  # **go 0x8000**

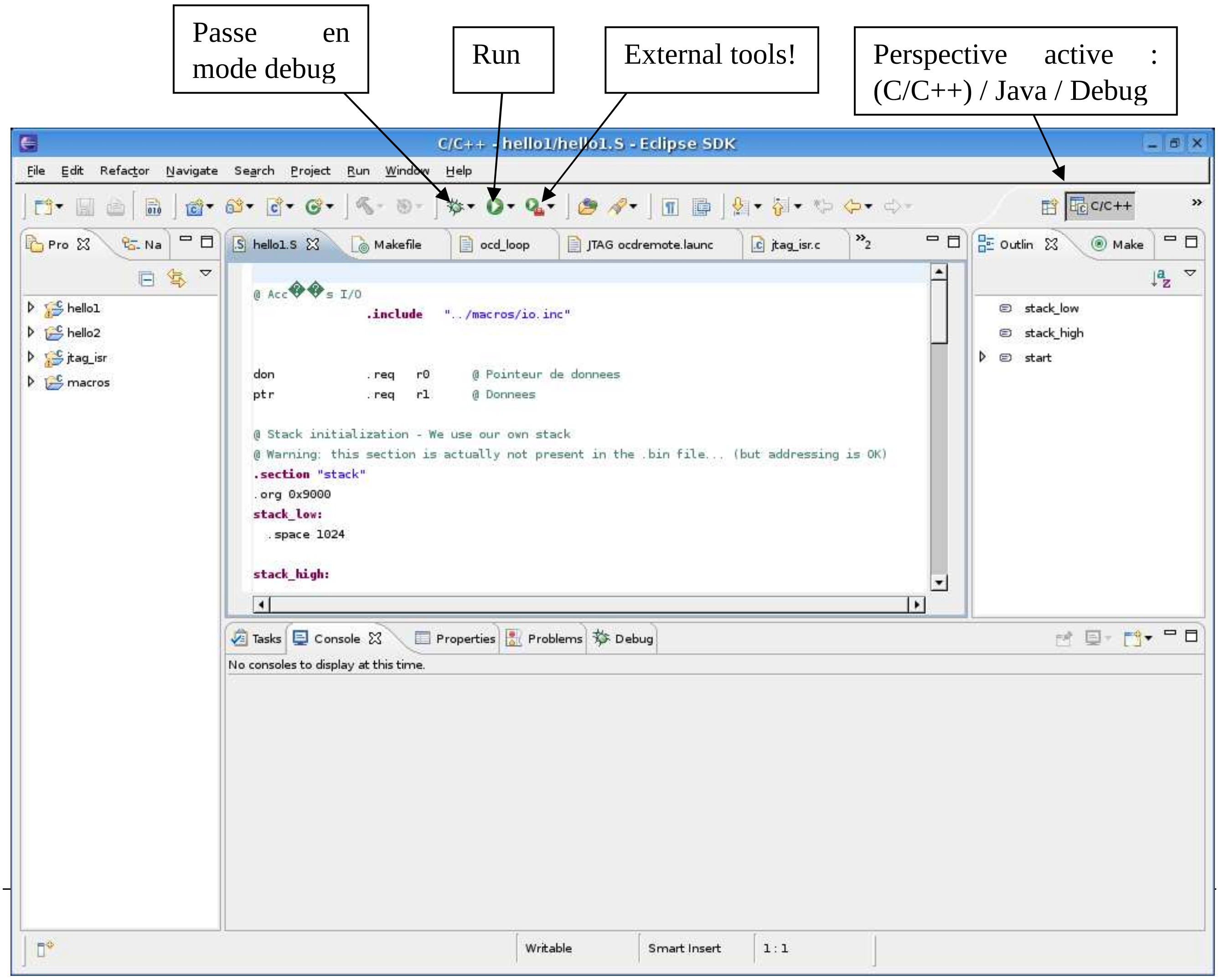
8. Debugg par JTAG USB

Pour réaliser du debug par JTAG USB sur la carte EMB du laboratoire, on utilise l'environnement de debugg *d'eclipse*.

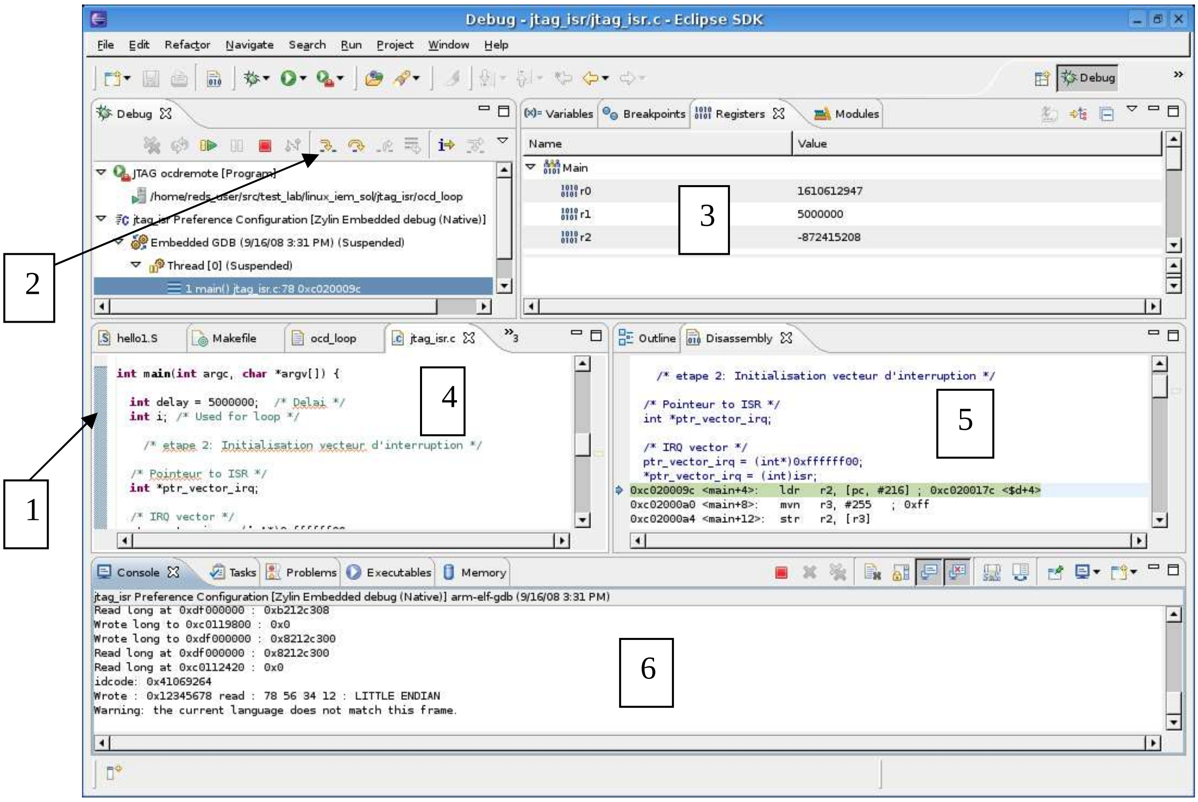


Choisissez-votre espace de travail!

Info utile:



Fenêtre debug:



1) Possibilité de mettre des breakpoints

2) Mode de debug (pas à pas, …)

3) Fenêtre des registres

4) et 5) Fenêtres contenant le code

6) Affichage de la console

9. Liens

Quelques lien qui peuvent vous êtes utiles :

- http://www.microcross.com/
- [http://www.**eclipse**.org/](http://www.eclipse.org/)